AMENDMENTS TO THE CLAIMS

Claim 1. (Currently amended) A method, comprising:

initiating compilation of a file in a processor-based system in advance of a request from a user to compile the file;

detecting the user request to compile the file; and

indicating a status of the compilation of the file in response to detecting the user request; wherein initiating compilation of the file comprises compiling the file in response to determining that the file has been modified.

Claim 2. (Original) The method of claim 1, wherein initiating compilation of the file comprises compiling the file including one or more code segments to produce an object code file.

Claim 3. (Original) The method of claim 2, wherein compiling the file comprises compiling one or more code segments in the file to produce an object code file, and further comprising linking the object code file to produce an executable file.

Claim 4. (Original) The method of claim 1, wherein indicating the status of the compilation of the file comprises at least one of indicating that the compilation was successful and indicating that the compilation was unsuccessful.

Claim 5. (Canceled)

Claim 6. (Currently amended) The method of claim ~~5~~ 1, wherein determining that the file has been modified comprises determining that the modified file has been saved to a storage unit.

Claim 7. (Currently amended) The method of claim 1, wherein the file includes one or more code segments, wherein initiating compilation of the file in response to determining that the file has been modified ~~further comprising~~ comprises:

~~determining that the file has been modified;~~

identifying the modified file in a work queue; and

initiating the compilation of the file based on the modified file being identified in the work queue.

Claim 8. (Original) The method of claim 1, wherein indicating the status of the compilation of the file comprises generating one or more files associated with the compilation of the file, storing the one or more generated files in a temporary location, and transferring the one or more files from the temporary location to a different location in response to detecting the user request.

Claim 9. (Currently amended) An article comprising one or more machine-readable storage media containing instructions that when executed enable a processor to:

initiate compiling of a file including one or more code segments;

detect a user request to compile the file; and

provide a result associated with the compiling in response to detecting the user request;

wherein the instructions when executed enable the processor to initiate compiling of the file based on determining that the file was modified.

Claim 10. (Original) The article of claim 9, wherein the instructions when executed enable the processor to display a message to a user indicating that one or more errors were detected during the compiling.

Claim 11. (Original) The article of claim 9, wherein the instructions when executed enable the processor to indicate to a user that the compiling was successful.

Claim 12. (Original) The article of claim 9, wherein the instructions when executed enable the processor to generate a file containing object code based on compiling the file and to store the object code file in a temporary location.

Claim 13. (Original) The article of claim 12, wherein the instructions when executed enable the processor to move the object code file from the temporary location into a product

location based on determining that the compiling of the file was successful and in response to detecting the user request.

Claim 14. (Canceled)

Claim 15. (Currently amended) The article of claim ~~14~~ 9, wherein the instructions that when executed enable the processor to initiate compiling of the file based on determining that the file was modified comprise instructions that when executed enable the processor to indicate in a work queue that the file has been modified and to initiate compiling of the file in response to detecting the indication.

Claim 16. (Currently amended) An apparatus, comprising:

means for initiating compilation of a file in a processor-based system in advance of a request from a user;

means for detecting the user request to compile the file; and

means for indicating a status of the compilation of the file in response to detecting the user request;

wherein the means for initiating compilation initiate compiling the file based on determining that the file was modified.

Claim 17. (Currently amended) An apparatus, comprising:

a storage unit having a file stored therein; and

a control unit communicatively coupled to the storage unit, the control unit adapted to:

initiate compilation of the file in advance of a request from a user to compile the file;

detect the user request to compile the file; and

indicate a status of the compilation of the file in response to detecting the user request;

wherein the control unit is adapted to compile the file in response to determining that the file has been modified.

Claim 18. (Original) The apparatus of claim 17, wherein the control unit is adapted to compile a file including one or more code segments to produce an object code file.

Claim 19. (Original) The apparatus of claim 18, wherein the control unit is adapted to link the object code file to produce an executable file.

Claim 20. (Original) The apparatus of claim 19, wherein the control unit is adapted to store the executable file in a temporary location and to transfer the executable file from the temporary location to a different location based on detecting the user request.

Claim 21. (Original) The apparatus of claim 18, wherein the control unit is adapted to at least one of indicate that the compilation was successful and indicate that the compilation was unsuccessful.

Claim 22. (Canceled)

Claim 23. (Currently amended) The apparatus of claim 17, wherein the control unit ~~is adapted~~ adaptation to compile the file in response to determining that the file has been modified comprises:

an adaptation to identify the modified file in a work queue; and

an adaptation to initiate the processing of the file based on the modified file being identified in the work queue.

Claim 24. (Original) A method, comprising:

identifying one or more source files that have been modified in a processor-based system;

initiating processing of at least a portion of the modified source files before receiving a request to process the modified files;

receiving the request to process at least one of the modified files; and

providing a status associated with the processing of the file in response to receiving the request.

Claim 25. (Original) The method of claim 24, wherein the processor-based system is adapted to execute an integrated development environment module, wherein identifying the one or more files comprises the integrated development environment module placing the one or more of the source files that have been modified in a queue.

Claim 26. (Original) The method of claim 25, wherein placing the one or more of the source files in the queue comprises placing at least one source file in the queue in response to a user saving the source file to a storage unit.

Claim 27. (Original) The method of claim 25, wherein placing the one or more of the source files in the queue comprises placing at least a portion of one source file in the queue in response to a user saving the source file to a storage unit using an editor and then exiting from the editor.

Claim 28. (Original) The method of claim 25, wherein placing the one or more of the source files in the queue comprises placing at least one source file in the queue in response to determining that a user desires to compile at least a portion of the source file as the source file is being edited.

Claim 29. (Original) The method of claim 25, wherein placing the one or more of the source files in the queue comprises placing at least one source file in the queue in response to determining that the source file includes at least one marker identifying a section of the source file that should be compiled, and wherein initiating processing of at least the portion of the one or more modified files comprises compiling the identified section of the source file.

Claim 30. (Original) The method of claim 25, wherein initiating the processing of the modified source files comprises causing a background thread to awaken in response to placing the one or more of the source files in the queue, where the background thread thereafter initiates processing of the source files.

Claim 31. (Original) The method of claim 25, wherein initiating the processing comprises initiating a build process to produce a software application and wherein receiving the request comprises receiving the request to building the software application.

Claim 32. (Original) The method of claim 25, wherein initiating the build process comprises performing compiling the modified source files to produce object code files and linking the object code files to produce executable files.

Claim 33. (Original) The method of claim 32, wherein the object code files and the executable files are stored in a first storage location.

Claim 34. (Original) The method of claim 32, further comprising suppressing at least one of an error and warning that is detected while compiling the modified source files.

Claim 35. (Original) The method of claim 32, wherein the object code files and the executable files are moved to a different storage location in response to detecting the request and in response to detecting no error or warning.

Claim 36. (Original) The method of claim 24, wherein identifying one or more source files comprises identifying the one or more source files based on a directed acyclic graph.

Claim 37. (Original) The method of claim 36, wherein the directed acyclic graph includes a list of dependent files, wherein identifying one or more source files comprises identifying at least one modified source file and another source file that is dependent on the modified source file using the directed acyclic graph.